Le système Sage

Nicolas M. Thiéry

Support repris de Franco Saliola, Florent Hivert, Dan Drake, William Stein, ...

IUT Orsay / Labo. de Mathématique, Univ. Paris XI

June 14, 2010

La mission de Sage

« Créer une alternative libre et viable à Maple^{TM}, Mathematica^{TM}, Magma^{TM} et MATLAB^{TM} [et MuPAD^{TM}]

...

et une communauté accueillante d'utilisateurs et de développeurs»

Une petite histoire

- *1999-2005.* William Stein écrit plus de 25,000 lignes de code Magma pour sa recherche. Il décide que est un mauvais investissement à long terme car il ne peux ni voir ni changer l'intérieur.

Une petite histoire

- *1999-2005.* William Stein écrit plus de 25,000 lignes de code Magma pour sa recherche. Il décide que est un mauvais investissement à long terme car il ne peux ni voir ni changer l'intérieur.
- *Fev. 2005.* SAGE-0.1, une bibliothèque Python liant ensemble PARI, Maxima, Python, Singular, GAP.
- *Fev. 2006.* SAGE-1.0 (& Sage Days 1 workshop)

Une petite histoire

- *1999-2005.* William Stein écrit plus de 25,000 lignes de code Magma pour sa recherche. Il décide que est un mauvais investissement à long terme car il ne peux ni voir ni changer l'intérieur.
- *Fev. 2005.* SAGE-0.1, une bibliothèque Python liant ensemble PARI, Maxima, Python, Singular, GAP.
- *Fev. 2006.* SAGE-1.0 (& Sage Days 1 workshop)
- *Sept. 2009.* Sage days 10 à Nancy (les premiers en Europe). 70 participants.

Une petite histoire

- *1999-2005.* William Stein écrit plus de 25,000 lignes de code Magma pour sa recherche. Il décide que est un mauvais investissement à long terme car il ne peux ni voir ni changer l'intérieur.
- *Fev. 2005.* SAGE-0.1, une bibliothèque Python liant ensemble PARI, Maxima, Python, Singular, GAP.
- *Fev. 2006.* SAGE-1.0 (& Sage Days 1 workshop)
- *Sept. 2009.* Sage days 10 à Nancy (les premiers en Europe). 70 participants.
- *Janv. 2010.* Sage Days 20 (les seconds en France), à Luminy 120 participants.

Une petite histoire

- *1999-2005.* William Stein écrit plus de 25,000 lignes de code Magma pour sa recherche. Il décide que est un mauvais investissement à long terme car il ne peux ni voir ni changer l'intérieur.
- *Fev. 2005.* SAGE-0.1, une bibliothèque Python liant ensemble PARI, Maxima, Python, Singular, GAP.
- *Fev. 2006.* SAGE-1.0 (& Sage Days 1 workshop)
- *Sept. 2009.* Sage days 10 à Nancy (les premiers en Europe). 70 participants.
- *Janv. 2010.* Sage Days 20 (les seconds en France), à Luminy 120 participants.
- *Version actuelle.* Sage-4.4.3.
- 10000 utilisateurs?

Une petite histoire

- *1999-2005.* William Stein écrit plus de 25,000 lignes de code Magma pour sa recherche. Il décide que est un mauvais investissement à long terme car il ne peux ni voir ni changer l'intérieur.
- *Fev. 2005.* SAGE-0.1, une bibliothèque Python liant ensemble PARI, Maxima, Python, Singular, GAP.
- *Fev. 2006.* SAGE-1.0 (& Sage Days 1 workshop)
- *Sept. 2009.* Sage days 10 à Nancy (les premiers en Europe). 70 participants.
- *Janv. 2010.* Sage Days 20 (les seconds en France), à Luminy 120 participants.
- *Version actuelle.* Sage-4.4.3.
- 10000 utilisateurs?
- *Financements.* NSF grants (postdoc, workshops, matériel); Google; Microsoft Research (port natif pour Windows); ANR, CNRS, nombreuses universités. . . .

Sage est très jeune!

Sage a:

- des bogues
- des incohérences
- des domaines vierges ou non documentées

Sage n'a pas:

- de support natif sous Windows (corrigé en 5.0!)
- de paquet fonctionnel sous Debian / Ubuntu / ...

Étude de cas: *-Combinat

`http://mupad-combinat.sf.net`

`http://combinat.sagemath.org`

- Boîte à outils pour l'exploration informatique en combinatoire
- 50+ articles de recherche
- Sponsorisé par NSF / ANR / Google
- MuPAD: 115k lignes de MuPAD, 15k lignes de C++, 32k lignes de tests, 600 pages de doc
- Sage: 300 tickets / 100k lignes intégrées dans Sage
- *Communauté*: Nicolas Borie, Daniel Bump, Jason Bandlow, Adrien Boussicault, Frédéric Chapoton, Vincent Delecroix, Tom Denton, François Descouens, Dan Drake, Teresa Gomez Diaz, Mike Hansen, Ralf Hemmecke, Florent Hivert, Brant Jones, Sébastien Labbé, Yann Laigle-Chapuy, Éric Laugerotte, Patrick Lemeur, Andrew Mathas, Gregg Musiker, Jean-Christophe Novelli, Janvier Nzeutchap, Steven Pon, Franco Saliola, Anne Schilling, Mark Shimozono, Lenny Tevlin, Nicolas Thiéry, Justin Walker, Qiang Wang, Mike Zabrocki, ...

Sage est une *distribution* de logiciels libres.

Sage est une *distribution* de logiciels libres.

Logiciels inclus dans Sage :

ATLAS	Automatically Tuned Linear Algebra Software
BLAS	Basic Fortan 77 linear algebra routines
Bzip2	High-quality data compressor
Cddlib	Double Description Method of Motzkin
Common Lisp	Multi-paradigm and general-purpose programming lang.
CVXOPT	Convex optimization, linear programming, least squares
Cython	C-Extensions for Python
F2c	Converts Fortran 77 to C code
Flint	Fast Library for Number Theory
FpLLL	Euclidian lattice reduction
FreeType	A Free, High-Quality, and Portable Font Engine

Sage est une *distribution* de logiciels libres.

Logiciels inclus dans Sage :

GAP	Groups, Algorithms, Programming
GD	Dynamic graphics generation tool
Genus2reduction	Curve data computation
Gfan	Gröbner fans and tropical varieties
Givaro	C++ library for arithmetic and algebra
GMP	GNU Multiple Precision Arithmetic Library
GMP-ECM	Elliptic Curve Method for Integer Factorization
GNU TLS	Secure networking
GSL	Gnu Scientific Library
JsMath	JavaScript implementation of LaTeX

Sage est une *distribution* de logiciels libres.

Logiciels inclus dans Sage :

IML	Integer Matrix Library
IPython	Interactive Python shell
LAPACK	Fortan 77 linear algebra library
Lcalc	L-functions calculator
Libgcrypt	General purpose cryptographic library
Libgpg-error	Common error values for GnuPG components
Linbox	C++ linear algebra library
Matplotlib	Python plotting library
Maxima	computer algebra system
Mercurial	Revision control system
MoinMoin	Wiki

Sage est une *distribution* de logiciels libres.

Logiciels inclus dans Sage :

MPFI	Multiple Precision Floating-point Interval library
MPFR	C library for multiple-precision floating-point computations
ECLib	Cremona's Programs for Elliptic curves
NetworkX	Graph theory
NTL	Number theory C++ library
Numpy	Numerical linear algebra
OpenCDK	Open Crypto Development Kit
PALP	A Package for Analyzing Lattice Polytopes
PARI/GP	Number theory calculator
Pexpect	Pseudo-tty control for Python
PNG	Bitmap image support

Sage est une *distribution* de logiciels libres.

Logiciels inclus dans Sage :

PolyBoRi	Polynomials Over Boolean Rings
PyCrypto	Python Cryptography Toolkit
Python	Interpreted language
Qd	Quad-double/Double-double Computation Package
R	Statistical Computing
Readline	Line-editing
Rpy	Python interface to R
Scipy	Python library for scientific computation
Singular	fast commutative and noncommutative algebra
Scons	Software construction tool
SQLite	Relation database

Sage est une *distribution* de logiciels libres.

Logiciels inclus dans Sage :

Sympow	L-function calculator
Symmetrica	Representation theory
Sympy	Python library for symbolic computation
Tachyon	lightweight 3d ray tracer
Termcap	for writing portable text mode applications
Twisted	Python networking library
Weave	Tools for including C/C++ code within Python
Zlib	Data compression library
ZODB	Object-oriented database

Sage est une *distribution* de logiciels libres.

Logiciels inclus dans Sage :

Sympow	L-function calculator
Symmetrica	Representation theory
Sympy	Python library for symbolic computation
Tachyon	lightweight 3d ray tracer
Termcap	for writing portable text mode applications
Twisted	Python networking library
Weave	Tools for including C/C++ code within Python
Zlib	Data compression library
ZODB	Object-oriented database

. . . ainsi que d'autres progiciels optionnels

Sage est une distribution de logiciels *mathématiques*.

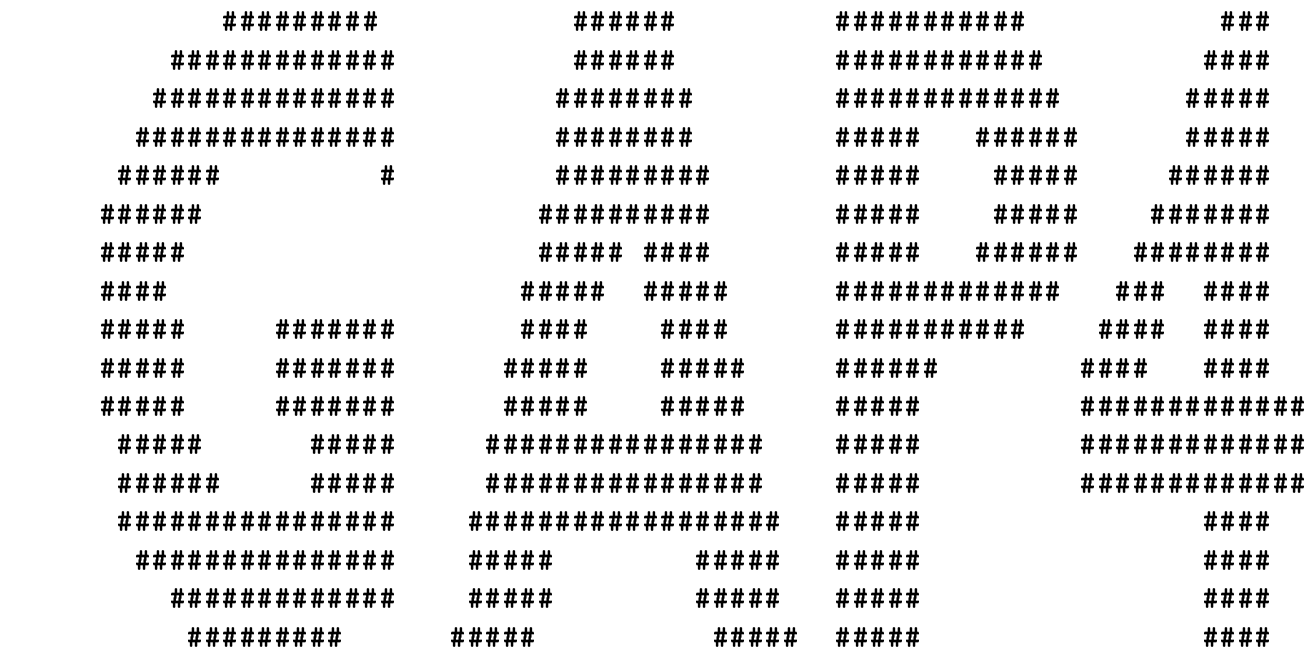
Algèbre	GAP, Maxima, Singular
Géométrie algébrique	Singular, Macaulay2 (optionnel)
Arithmétique en précision arbitraire	GMP, MPFR, MPFI, NTL, ...
Géométrie arithmétique	PARI, NTL, mwrank, ecm, ...
Calcul	Maxima, Sympy
Combinatoire	Symmetrica, MuPAD-Combinat*
Algèbre linéaire exacte	Linbox, IML
Théorie des graphes	NetworkX
Théorie des groupes	GAP
Algèbre linéaire numérique	GSL, Scipy, Numpy

Sage est une distribution de logiciels *mathématiques*.

Algèbre	GAP, Maxima, Singular
Géométrie algébrique	Singular, Macaulay2 (optionnel)
Arithmétique en précision arbitraire	GMP, MPFR, MPFI, NTL, ...
Géométrie arithmétique	PARI, NTL, mwrank, ecm, ...
Calcul	Maxima, Sympy
Combinatoire	Symmetrica, MuPAD-Combinat*
Algèbre linéaire exacte	Linbox, IML
Théorie des graphes	NetworkX
Théorie des groupes	GAP
Algèbre linéaire numérique	GSL, Scipy, Numpy

... et bien d'autres!

```
> sage -gap
```



```
    Information at:  http://www.gap-system.org
    Try '?help' for help. See also  '?copyright' and  '?authors'

   Loading the library. Please be patient, this may take a while.
GAP4, Version: 4.4.10 of 02-Oct-2007, x86_64-unknown-linux-gnu-gcc

gap>
```

```
> sage -singular

                     SINGULAR                                 /  Development
 A Computer Algebra System for Polynomial Computations       /   version 3-1-0
                                                            0<
     by: G.-M. Greuel, G. Pfister, H. Schoenemann             \   Mar 2009
FB Mathematik der Universitaet, D-67653 Kaiserslautern         \
>
```

```
> sage -maxima

Maxima 5.16.3 http://maxima.sourceforge.net
Using Lisp ECL 9.4.1
Distributed under the GNU Public License. See the file COPYING.
Dedicated to the memory of William Schelter.
The function bug_report() provides bug reporting information.
(%i1)
```

```
> sage -gp

                GP/PARI CALCULATOR Version 2.3.3 (released)
           amd64 running linux (x86-64/GMP-4.2.1 kernel) 64-bit version
            compiled: Jul 10 2009, gcc-4.3.2 (Ubuntu 4.3.2-1ubuntu12)
                (readline v5.2 enabled, extended help available)

                        Copyright (C) 2000-2006 The PARI Group

PARI/GP is free software, covered by the GNU General Public License, and
comes WITHOUT ANY WARRANTY WHATSOEVER.

Type ? for help, \q to quit.
Type ?12 for how to get moral (and possibly technical) support.

parisize = 8000000, primelimit = 500000
?
```

```
> sage -R

R version 2.6.1 (2007-11-26)
Copyright (C) 2007 The R Foundation for Statistical Computing
ISBN 3-900051-07-0

R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.

  Natural language support but running in an English locale

R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.

Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

>
```

Sage *combine* la puissance de multiples logiciels

Sage contient une *bibliothèque native* :

- 500k lignes
- 2500 classes
- 27058 fonctions
- contenu mathématique: catégories, combinatoire, graphes, théorie des nombres, ...

La langage de programmation de Sage est *Python*

Python est un langage de programmation très puissant, moderne, et interprété.

La langage de programmation de Sage est *Python*

Python est un langage de programmation très puissant, moderne, et interprété.

- *interprété* facile à utiliser comme une calculette ...

```
python: x = 17
python: x
17
python: 3*x
51
```

La langage de programmation de Sage est *Python*

Python est un langage de programmation très puissant, moderne, et interprété.

- *interprété* facile à utiliser comme une calculette ...

```
python: x = 17
python: x
17
python: 3*x
51
```

- Il est facile à apprendre; beaucoup de documentation libre:
  - Plongez au coeur de Python (http://diveintopython.adrahon.org)
  - Tutoriel Python (http://www.ceramiko.ch/python/main.html)
  - Tutoriel Sage (http://wiki.sagemath.org/i18n/French)

Le langage de programmation de Sage est *Python*

- Facile à lire et à écrire:

math: $\left\{ 17x \;\middle|\; x \in \{0, 1, \ldots, 9\} \text{ and } x \text{ is odd} \right\}$

```
python: [17*x for x in range(10) if x%2 == 1]
```

Le langage de programmation de Sage est *Python*

- Facile à lire et à écrire:

math: $\left\{ 17x \;\middle|\; x \in \{0, 1, \ldots, 9\} \text{ and } x \text{ is odd} \right\}$

```
python: [17*x for x in range(10) if x%2 == 1]
```

- Beaucoup de *bibliothèques* pour Python: base de données, graphiques, réseau, ...

Le langage de programmation de Sage est *Python*

- Facile à lire et à écrire:

$$\text{math: } \left\{ 17x \;\middle|\; x \in \{0, 1, \ldots, 9\} \text{ and } x \text{ is odd} \right\}$$

```
python: [17*x for x in range(10) if x%2 == 1]
```

- Beaucoup de *bibliothèques* pour Python: base de données, graphiques, réseau, ...
- Facile d'interfacer des *bibliothèques C/C++* avec Python.

Le langage de programmation de Sage est *Python*

- Facile à lire et à écrire:

$$\text{math: } \left\{ 17x \;\middle|\; x \in \{0, 1, \ldots, 9\} \text{ and } x \text{ is odd} \right\}$$

```
python: [17*x for x in range(10) if x%2 == 1]
```

- Beaucoup de *bibliothèques* pour Python: base de données, graphiques, réseau, ...
- Facile d'interfacer des *bibliothèques C/C++* avec Python.
- *Cython*: code Python $\longmapsto$ code C.

La langage de programmation de Sage est *Python*

"Google has made no secret of the fact they use Python a lot for a number of internal projects. Even knowing that, once I was an employee, I was amazed at how much Python code there actually is in the Google source code system."

— *Guido van Rossum*

(créateur de Python)

Sage est entièrement libre

"You can read Sylow's Theorem and its proof in Huppert's book in the library . . . then you can use Sylow's Theorem for the rest of your life free of charge, but for many computer algebra systems license fees have to be paid regularly . . . .

With this situation two of the most basic rules of conduct in mathematics are violated: In mathematics information is passed on free of charge and everything is laid open for checking."

— *J. Neubüser (1993)*

(started GAP in 1986)

Liberté d'exécution

En tout lieu, pour tout usage, dans les seules limites de la loi.

Liberté d'exécution

En tout lieu, pour tout usage, dans les seules limites de la loi.

Liberté de redistribution de copies

Inclue la liberté de vendre des copies!

Liberté d'exécution

En tout lieu, pour tout usage, dans les seules limites de la loi.

Liberté de redistribution de copies

Inclue la liberté de vendre des copies!

Intérêt:

- Simplicité administrative et technique

Liberté d'exécution

En tout lieu, pour tout usage, dans les seules limites de la loi.

Liberté de redistribution de copies

Inclue la liberté de vendre des copies!

Intérêt:

- Simplicité administrative et technique
- Utilisation par les étudiants chez eux

Liberté d'exécution

En tout lieu, pour tout usage, dans les seules limites de la loi.

Liberté de redistribution de copies

Inclue la liberté de vendre des copies!

Intérêt:

- Simplicité administrative et technique
- Utilisation par les étudiants chez eux
- Calcul distant, déploiement à large échelle

Liberté d'exécution

En tout lieu, pour tout usage, dans les seules limites de la loi.

Liberté de redistribution de copies

Inclue la liberté de vendre des copies!

Intérêt:

- Simplicité administrative et technique
- Utilisation par les étudiants chez eux
- Calcul distant, déploiement à large échelle
- Partage de vos programmes, feuilles de travail

Liberté d'exécution

En tout lieu, pour tout usage, dans les seules limites de la loi.

Liberté de redistribution de copies

Inclue la liberté de vendre des copies!

Intérêt:

- Simplicité administrative et technique
- Utilisation par les étudiants chez eux
- Calcul distant, déploiement à large échelle
- Partage de vos programmes, feuilles de travail

- Non discrimination
- Libre accès aux non académiques
- Pays en voie de développement

Liberté d'étude du fonctionnement

Logiciel *ouvert*: accès au code source

Liberté d'étude du fonctionnement

Logiciel *ouvert*: accès au code source

Intérêt:

- Enseignement

Liberté d'étude du fonctionnement

Logiciel *ouvert*: accès au code source

Intérêt:

- Enseignement
- Diffusion de la science

Liberté d'étude du fonctionnement

Logiciel *ouvert*: accès au code source

Intérêt:

- Enseignement
- Diffusion de la science
- D'autant plus utile que le code est *expressif*

Liberté d'étude du fonctionnement

Logiciel *ouvert*: accès au code source

Intérêt:

- Enseignement
- Diffusion de la science
- D'autant plus utile que le code est *expressif*
- Reproductibilité des résultats scientifiques

Liberté d'étude du fonctionnement

Logiciel *ouvert*: accès au code source

Intérêt:

- Enseignement
- Diffusion de la science
- D'autant plus utile que le code est *expressif*
- Reproductibilité des résultats scientifiques
- Vérification des démonstrations

Liberté d'étude du fonctionnement

Logiciel *ouvert*: accès au code source

Intérêt:

- Enseignement
- Diffusion de la science
- D'autant plus utile que le code est *expressif*
- Reproductibilité des résultats scientifiques
- Vérification des démonstrations
- Maîtrise des hypothèses, modèles et algorithmes

Liberté d'étude du fonctionnement

Logiciel *ouvert*: accès au code source

Intérêt:

- Enseignement
- Diffusion de la science
- D'autant plus utile que le code est *expressif*
- Reproductibilité des résultats scientifiques
- Vérification des démonstrations
- Maîtrise des hypothèses, modèles et algorithmes
- Analyse de boggues ou de comportements surprenants

Liberté d'améliorer et de publier ses améliorations

Liberté d'améliorer et de publier ses améliorations

Intérêt:

- Adaptation aux besoins locaux (langue, terminologie métier)
- Développements spécifiques

Liberté d'améliorer et de publier ses améliorations

Intérêt:

- Adaptation aux besoins locaux (langue, terminologie métier)
- Développements spécifiques
- Corrections de boggues

Liberté d'améliorer et de publier ses améliorations

Intérêt:

- Adaptation aux besoins locaux (langue, terminologie métier)
- Développements spécifiques
- Corrections de boggues

- Autonomie et pouvoir potentiel

Liberté d'améliorer et de publier ses améliorations

Intérêt:

- Adaptation aux besoins locaux (langue, terminologie métier)
- Développements spécifiques
- Corrections de boggues

- Autonomie et pouvoir potentiel
- Mutualisation des efforts
- Importance des communautés

La communauté Sage dans le monde

Il y a actuellement 184 contributeurs à 118 endroits

La communauté Sage

sites web

`http://www.sagemath.org/`

`http://www.sagemath.fr/`

mailing lists

`sage-devel`: développement

`sage-windows`: port vers Windows

`sage-release` gestion des distributions

`sage-algebra`: algèbre

`sage-combinat-devel`: combinatoire

`sage-finance` finance

`sage-nt` théorie des nombres

`sage-grid` pour le calcul scientifique en grille

`sage-edu` pour l'enseignement

`sagemath-edu` pour l'enseignement (en Français)

irc-channel

`#sage-devel` on `freenode.net`

Journées Sage!

- Ateliers intensifs pour attirer de nouveaux utilisateurs et développeurs
- Conférences d'introduction, tutoriaux
- Conférences d'experts
- Discussions de conceptions
- Coding sprints!
- 20-120 personnes

Journées Sage en 2010

- Sage Days 19: Seattle, WA (January 2010)
- Sage Days 20: Marseille (February 2010)
- Sage Days 20.25: Montreal (March 2010)
- Sage Days 20.5: Fields Institute (May 2010)
- Sage Days 21: Seattle, WA (June 2010)
- Sage-Combinat/Chevie: France (June 2010)
- Sage Days 22: Berkeley, CA (July 2010)
- Sage Days 23: Leiden, Netherlands (July 2010)
- Sage Days 24: Linz, Austria (July 2010)
- Sage Days 25: Mumbai, India (August 2010)
- Sage Days 26: Kaiserslautern, Germany (August 2010)

Objectif: code, tests et documentation de haute qualité

Tout nouveau code est:

- rigoureusement testé
- bien documenté
- revu par les pairs

```
> sage -coverage
...
Overall weighted coverage score:  81.1%
Total number of functions:  27058
We need 1061 more function to get to 85% coverage.
We need 2414 more function to get to 90% coverage.
We need 3767 more function to get to 95% coverage.
```

Cycle de développement

`http://trac.sagemath.org/`

Tickets (exemples: #8154, #8890)

Patchs

Nouvelles versions toutes les 2–3 semaines

«Calculer, avec Sage »

Alexandre Casamayou, Guillaume Connan
Thierry Dumont, Laurent Fousse
François Maltey, Matthias Meulien
Marc Mezzarobba, Clément Pernet
Nicolas Thiéry, Paul Zimmermann

Un livre libre, disponible sur le web en juillet

Suite de «Calcul formel, mode d'emploi»
(Dumas, Gomez, Salvy, Zimmermann)